Problem A. Cool, It's Yesterday Four Times More

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

After the great success in 2018, 2019, 2020, 2021 and 2022, Nanjing University of Aeronautics and Astronautics (NUAA) will host the *International Collegiate Programming Contest* (ICPC) Nanjing regional for the sixth time in a row.

Team ***Power of Two*** and team ***Three Hold Two*** won the champion title for Tsinghua University in 2018 and 2019. In 2020, 2021 and 2022, team ***Inverted Cross*** from Peking University won the three-peat champion titles. This year, there are around 330 teams participating in the contest. There are at most 33 gold medals, 66 silver medals and 99 bronze medals that will be awarded (note that these numbers are for reference only). We are looking forward to seeing participants' outstanding performance!

What's even better is that, as the pandemic has come to an end, we can finally gather in Nanjing to participate in this wonderful contest. We'd like to be grateful for the hard work done by all staff and volunteers for this contest. Thank you all for your great contribution to this contest!



The 2018 ICPC Asia Nanjing Regional Contest

In the 2018 contest, problem K, ***Kangaroo Puzzle***, requires the contestants to construct an operation sequence for the game:

> The puzzle is a grid with n rows and m columns ($1 \le n, m \le 20$) and there are some (at least 2) kangaroos standing in the puzzle. The player's goal is to control them to get together. There are some walls in some cells and the kangaroos cannot enter the cells with walls. The other cells are empty. The kangaroos can move from an empty cell to an adjacent empty cell in four directions: up, down, left, and right.
>
> There is exactly one kangaroo in every empty cell in the beginning and the player can control the kangaroos by pressing the button U, D, L, R on the keyboard. The kangaroos will move simultaneously according to the button you press.
>
> The contestant needs to construct an operating sequence of at most 5×10^4 steps consisting of U, D, L, R only to achieve the goal.

In the 2020 contest, problem A, ***Ah, It's Yesterday Once More***, requires the contestants to construct an input map to hack the following code of the problem described before:

```
#include <bits/stdc++.h>
using namespace std;
string s = "UDLR";
int main()
{
  srand(time(NULL));
  for (int i = 1; i <= 50000; i++) putchar(s[rand() % 4]);
  return 0;
}
```

In the 2021 contest, problem A, ***Oops, It's Yesterday Twice More***, also requires the contestants to construct an operation sequence for the game:

> This time, every cell in the grid stands exactly one kangaroo. You need to construct an operating sequence consisting only of characters 'U', 'D', 'L', and 'R'. After applying it, you must make sure every kangaroo will gather at the specific cell (a, b). The length of the operating sequence cannot exceed $3(n-1)$. As always, the kangaroos will move simultaneously according to the operation you command.

In the 2022 contest, problem A, ***Stop, Yesterday Please No More***, asks the contestants to solve the following counting problem:

> This time, every cell (except one which is a hole) in the grid stands exactly one kangaroo. The operating sequence is given and all kangaroos stepping out of the grid or onto the hole will be removed. Given the number of kangaroos remaining after all operations, count the number of positions which might be the hole.

Now, in the 2023 contest, the kangaroo problem is back again! We don't know why problem setters are so obsessed with kangaroos but the problem is as follows:

You are given a grid with n rows and m columns. Each cell is either a hole or empty. In each empty cell stands exactly one kangaroo.

Similarly, the kangaroos are controlled by pressing the button U, D, L, R on the keyboard. All kangaroos will move simultaneously according to the button pressed. Specifically, for any kangaroo located in the cell on the i-th row and the j-th column, indicated by (i, j):

1. Button U: it will move to $(i-1, j)$.
2. Button D: it will move to $(i+1, j)$.
3. Button L: it will move to $(i, j-1)$.
4. Button R: it will move to $(i, j+1)$.

If a kangaroo steps onto a hole or steps out of the grid, it will be removed from the grid. If after applying a sequence of operations (possibly an empty sequence) there is exactly one kangaroo remaining on the grid, that kangaroo becomes the winner.

The problem is: for each kangaroo, determine if there exists a sequence of operations to make it the winner. Output the total number of kangaroos which are possible to become the winner.

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line contains two integers n and m ($1 \le n, m \le 10^3$, $1 \le n \times m \le 10^3$) indicating the number of rows and columns of the grid.

For the following n lines, the i-th line contains a string $s_{i,1}s_{i,2}\cdots s_{i,m}$ of length m where each character is either '.' (dot, ascii: 46) or 'O' (capitalized letter, ascii: 79). If $s_{i,j}$ is a '.' then grid (i, j) is empty; If $s_{i,j}$ is a 'O' then grid (i, j) is a hole.

It's guaranteed that the sum of $n \times m$ of all test cases will not exceed 5×10^3.

Output

For each test case output one line containing one integer indicating the number of kangaroos for which there exists a sequence of operations to make it the winner.

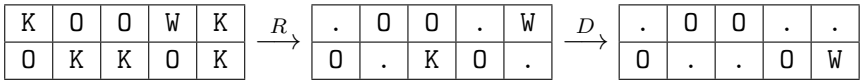
Example

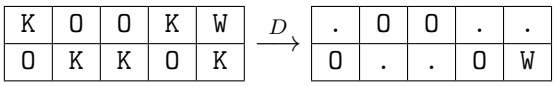
standard input	standard output
4 2 5 .OO.. O..O. 1 3 O.O 1 3 .O. 2 3 OOO OOO	3 1 0 0

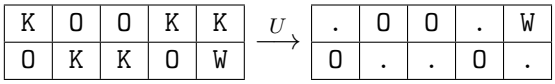
Note

The sample test cases are explained below. We use 'W' to indicate the kangaroo which later becomes winner and 'K' to indicate the other kangaroos.

For the first sample test case, kangaroos initially located at $(1, 4)$, $(1, 5)$ and $(2, 5)$ may become winners. Possible sequences of operations are shown as follows:







For the second sample test case, as there is only one kangaroo, no operation is needed for it to become the winner.

For the third sample test case, as any operation will remove the two kangaroos at the same time, there is no possible winner.

For the fourth sample test case, as there is no kangaroo, there is no possible winner.